



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/786,004

02/26/2004

Russell Norman Owen

13210-16

3548

1059 7590 12/27/2007

BERESKIN AND PARR  
40 KING STREET WEST  
BOX 401  
TORONTO, ON M5H 3Y2  
CANADA

EXAMINER

GUPTA, MUKTESH G

ART UNIT

PAPER NUMBER

4121

MAIL DATE

DELIVERY MODE

12/27/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/786,004	<b>Applicant(s)</b> OWEN ET AL.	
	<b>Examiner</b> Muktesh G. Gupta	<b>Art Unit</b> 4121	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 26 February 2004.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-50 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-50 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)            | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>03/05/2007, 12/20/2006, 11/25/2005, 09/02/2004</u> .          | 6) <input type="checkbox"/> Other: _____                          |



### DETAILED ACTION

1. **Claims 1-50** have been examined and are pending.

#### ***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. **Claims 1-50** rejected under 35 U.S.C. 102(e) as being anticipated by US Publication No. 20050044197 to Lai, Ray Y., (hereinafter “Lai”).

**As to Claims 1, 24 and 50**, Lai anticipates method, web services program, stored on computer-readable medium, of processing a web service description so that said web service description is adapted for use with mobile devices, said web service description comprising a plurality of web service description elements, said method comprising the steps of (as stated in par. 0213, lines 1, par. 0412, lines 1-2 Service Requester or consumer accessing web services from variety of **mobile devices** which are defined by web service provider in service **description, configuration information, network connection**, and **service end-points** in **WSDL-Web Services Description Language**):

a) receiving a web service description, wherein said web service description defines an interface to a web service (as stated in par. 0262, lines 6-8, web Service Requester receives WSDL-Web Services Description Language in the XML structure which **defines** the **interface** to the web service);

b) creating at least one accelerator output file from said web service description, wherein said at least one accelerator output file is adapted for processing by a mobile device (as stated in par. 0536, lines 6-8, par. 0413, lines 1-5, dynamic web service look-up is beneficial to managing business services within a large user community, content providers and mobile phone subscribers, where business **web services** and **content** are **added or modified** in real-time by the Service Delivery component's, **controller servlet** that can handle service requests from the Service Requester's **mobile devices** or browser);

and c) transmitting said at least one accelerator output file to said mobile device, wherein said at least one accelerator output file facilitates invocations of said web service by said mobile device (as stated in par. 0414, lines 1-9 and par. 0415, lines 1-7, web **service data** returned from the RPC router servlet is captured by the service proxy in XML format. The presentation manager may **reformat the data and transcode into HTML, or WML using XSL if applicable**. This enables the Service Requester to **view** in a **format** that is **displayable** on any **mobile device** or browser).

**As to Claims 2 and 25**, Lai anticipates method and web services program of claims 1 and 24, wherein said web service description is in Web Service Description Language (as stated in par. 0262, lines 4-9, **Web Services** technology **uses** the **Web Services Description Language ( WSDL)** in the XML structure to define the interfaces, network connection, and service end-points).

**As to Claims 3 and 26**, Lai anticipates method and web services program of claims 1 and 24, wherein invocations of the web service by the mobile device are performed in accordance with a Simple Object Access Protocol (as stated in par. 0265, lines 1-4, architecture framework of **Web Services** is based on open standards technology, J2EETM, XML, **SOAP, (Simple Object Access Protocol)** and UDDI).

**As to Claims 4 and 27**, Lai anticipates method and web services program of claims 1 and 24, wherein said creating step comprises at least one optimizing step at which said web service description is optimized for said mobile device, wherein said at least one accelerator output file is an optimized web service description (as stated in par. 0585, lines 1-4, par. 0586, lines 1-4, par. 0587, lines 1-4 and par. 0588, lines 1-4, web service description is **optimized** with administering Multiple Servlet Engines, different configurations for different servlet engines in the server.xml **file**, different services and resources mapping. Vertical scaling **optimizes** existing computing capacity by running multiple

instances of the **SOAP servlet** engines. Multiple servlet engines support application partitioning to preferably provide better **performance** and **scalability**).

**As to Claims 5 and 28**, Lai anticipates method and web services program of claims 4 and 27, wherein said at least one optimizing step comprises resolving symbolic references in said web service description such that said at least one accelerator output file is adapted to be parsed by said mobile device in one pass (as stated in par. 1459, lines 1-3 and par. 1459, lines 1-13, ProfileServlet parses the SOAP request using the JAXM/TSIK Message Provider for optimizing and resolving **symbolic references** for **single sign-on** by web service requester using **mobile device**).

**As to Claims 6 and 29**, Lai anticipates method and web services program of claims 4 and 27, wherein each of a subset of said plurality of web service description elements is associated with transport protocols not supported by said mobile device, and wherein said at least one optimizing step comprises identifying said subset, and excluding said subset from said at least one accelerator output file created at said creating step (as stated in par. 0972, lines 5-9, Web Services technology, middleware integration options use SOAP over HTTPS as the **transport** which **decouples (excludes)** the middleware from the **data transport integration**, making interoperability **easier (optimized)**).

**As to Claims 7 and 30**, Lai anticipates method and web services program of claims 4 and 27, wherein said at least one optimizing step comprises modifying one or more names associated with each of one or more web service description elements (as stated in par. 0865, lines 1-9, Java transcoding tools allow platform environment parameters, **dataset names** to be **changed (modifying)** intelligently to the new target environment).

**As to Claims 8 and 31**, Lai anticipates method and web services program of claims 4 and 27, further comprising validating said at least one accelerator output file (as stated in par. 0331, lines 1-3 and par. 0332, lines 1-3, Web Services technology Compose or transform XML documents, **validate** their well-formedness using SAX or DOM with XML Schema and if necessary, **transform** one format to another format or XML variant or render XML documents into **different devices** or multichannels).

**As to Claim 9**, Lai anticipates method of claim 4, further comprising processing said at least one accelerator output file by identifying web service description elements that define inputs to said web service, a destination, and a format for said inputs from said optimized web service description (as stated in par. 0224, lines 1-6, web services technology is easy and quick to deploy and integration framework support different protocols and message **formats**,



including a variety of industry standards and platforms which are **identified**, **integrated** and **delivered** in minimal time frame).

**As to Claim 10**, Lai anticipates method of claim 9, further comprising invoking said web service by transmitting input data to said destination in said format (as stated in par. 1402, lines 1-16, Open Standards messaging protocols are used for the interaction between different components of web services, Clients, Control Servlet, Reference Data, Service Provider and the Registry Server. Control Servlet handles Presentation-Tier requests, acts as a SOAP client to look up the web services dynamically using JAXR, retrieves **reference data** using JAXM and **invokes** remote **web services** using JAX-RPC).

**As to Claim 11**, Lai anticipates method of claim 10, further comprising receiving output data from said web service in response to said invoking step (as stated in par. 0415, lines 1-6, business **data returned** from the RPC router servlet is captured by the service proxy in XML format. The presentation manager **reformats** the data and transcode into HTML, or WML using XSL if applicable. This enables the Web Service Requester to **view** in a **format** that is **displayable** on any **mobile device** or browser).

**As to Claims 12 and 32**, Lai anticipates method and web services program of claims 4 and 27, wherein said at least one optimizing step comprises

extracting invocation information from said web service description, and storing said invocation information (as stated in par. 0898, lines 1-11, In this stage, a new CRM business ***data model (database storage)*** is defined and customized from the customer ***information extracted*** from existing delivery channels such as ATM channel and teller platform. The data extraction, one-way data synchronization using the existing middleware or messaging infrastructure is achieved by implementing ***web services***).

**As to Claims 13 and 33**, Lai anticipates method and web services program of claims 12 and 32, further comprising processing said at least one accelerator output file by identifying web service description elements that define inputs to said web service and obtaining operation parameters based on said inputs (as stated in par. 0763, 0764, lines 1-3, front ***controller servlet*** initiates EPI, ECI calls, the objective is to invoke a CICS mainframe transaction ID and ***return*** the response for terminal ***operation parameters*** data).

**As to Claims 14 and 34**, Lai anticipates method and web services program of claims 13 and 33, further comprising generating input data by combining said operation parameters with said invocation information (as stated in par. 0777, lines 1-13, CICS resource requests from the CICS socket, web attach program and the alias program CWBA handle the ***conversion*** of the ***code page*** from ANSI to EBCDIC. The process ***converts input data*** from the request to ASCII,

which may then be converted to EBCDIC. The analyzer look up the alias name, converter, user ID, and user program before passing the CICS request to the Alias program CWBA and functions as a converter, which **encodes** and **decodes** CICS **data**.

**As to Claims 15 and 35**, Lai anticipates method and web services program of claims 14 and 34, further comprising invoking said web service by transmitting said input data to said web service (as stated in par. 0777, lines 1-13, Alias program then **invokes** the appropriate **user application** program to process the CICS requests. The Alias program receives the presentation logic decoded in CICS **format**. The Alias program sends the CICS results in encoded data that can be handled by the presentation logic such as **HTML Web pages** displayed on the **mobile device**).

**As to Claims 16 and 36**, Lai anticipates method and web services program of claims 15 and 35, further comprising receiving output data from said web service in response to said invoking step (as stated in par. 0777, lines 1-13, The Alias program receives the presentation logic decoded in CICS **format**. The Alias program **sends** the CICS **results** in **encoded data** that can be **handled** by the **presentation logic** such as **HTML Web pages** displayed on the **mobile device**).

**As to Claims 17 and 41**, Lai anticipates method and web services program of claims 1 and 24, wherein said at least one accelerator output file comprises code adapted for execution on said mobile device, for obtaining input data used to invoke said web service, and for invoking said web service using said input data. (as stated in par. 0892, lines 1-15 and par. 0729, lines 4-9, J2EE-compliant applications web server provide Web and EJB containers to develop and **execute** Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML-related Java APIs to develop XML-based **transforming** and **Web Services** through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be **executed** on the **mobile platform** devices. **Web Services** requester may start with one single coarse-grained Web Service, where different services are **invoked** by passing **parameters** in the Web Service call either an XML-RPC or an XML message).

**As to Claims 18 and 42**, Lai anticipates method and web services program of claims 17 and 41, wherein said creating step comprises the following substeps:

a) identifying web service description elements that define inputs to said web service from said web service description (as stated in par. 0301, lines 1-7, web service technologies implement service proxy, JAXR, **search engines** for searching registry with their search algorithm, combination of logical operators for **identifying elements** that define inputs to web service);

b) producing first instructions for generating a user interface to prompt a user for one or more of said inputs to said web service (as stated in par. 0302, lines 1-7 and par. 0310, lines 1-7 JAXR pluggable provider **class** is able to use any capability-specific **interfaces** such as ebXML provider or UDDI provider that is specific to a particular Service Registry platform and require to **input** to many **data entry** screens for creating a service record and access registry content, objects, catalogs, specifications, WSDL documents, standards and XML schema);

c) producing second instructions for obtaining input data associated with said one or more inputs (as stated in par. 0303, lines 1-7, APIs or servlets are used to access their registry contents. When Java API for Registries JAXR is **used**, the **same** Java API may be used with different registry properties. Property file stores the specific registry servlet or URIs, Universal Resource Identifier, or the service endpoints URLs);

d) identifying web service description elements that define a destination and a format for said inputs to said web service (as stated in par. 0240, lines 1-7, web service providers use different **messaging protocols** and data **formats**, they use SOAP or ebXML messaging to exchange **documents or instructions**. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a **format** that can be understood by the back-end ERP or legacy systems);

and e) producing third instructions for invoking said web service by transmitting said input data to said destination in said format. (as stated in par. 0422, lines 1-7, controller servlet are **documents or instructions** encapsulated in XML structure for the **Integration** Tier to hosts the integration components such as **messaging bus**, **gateways** such as Host Gateway for legacy mainframe systems, and security components such as Directory Server and Policy Server. The Host Gateway provides a channel to **invoke** applications running on legacy mainframes for appropriate **web services**).

**As to Claims 19 and 44**, Lai anticipates method and web services program of claims 18 and 43, wherein said creating step further comprises the substeps of:

f) identifying web service description elements that define outputs from said web service in response to invocations of said web service and a format for said outputs from said web service description (as stated in par. 0422, lines 1-18, controller servlet are **documents or instructions** encapsulated in XML structure for the Integration Tier to hosts the integration components such as messaging bus, gateways such as Host Gateway for **other web services**. The Directory Server provides **enterprise-level** authentication for **web services**. The Policy Server stores access rights and policies that govern the access level of **each web service component** or system by users and by roles. These security components **span two or more tiers**);

and g) producing fourth instructions for receiving output data in said format from said web service (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, **messaging calls** and **data returned** from the RPC router servlet is **captured** by the service proxy from the Resource Tier which hosts all data stores residing on a database server running a relational database, data warehouse, Enterprise Resource Planning (ERP) systems, and legacy mainframe applications).

**As to Claims 20 and 45**, Lai anticipates method and web services program of claims 19 and 44, wherein said creating step further comprises the substep of:

h) producing fifth instructions for outputting output data received from said web service to said user (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager **reformats** the **data** and **transcode** into **HTML**, or **WML** using XSL if applicable. This enables the Service Requester (**user**) to view in a **format** that is displayable on any **mobile device or browser**).

**As to Claims 21 and 48**, Lai anticipates method and web services program of claims 17 and 35, wherein said creating step further comprises compiling instructions produced at said creating step into said code (as stated in par. 0872, lines 1-6, web services technologies implement cross-**compiling** of the source **program** language structure such as COBOL to a target program language structure such as **Java byte-code** using an intelligent language cross-compiler without changing the application program logic, structured **procedures** and

**instructions** may then be transcoded into Java beans or EJBs for optimizing web services).

**As to Claims 22 and 49**, Lai anticipates method and web services program of claims 21 and 35, wherein said code represents an executable Java application (as stated in par. 0872, lines 1-6, web services technologies implement cross-**compiling** of the source **program** language structure such as COBOL to a target program language structure such as **Java byte-code** using an intelligent language cross-compiler without changing the application program logic, structured **procedures** and **instructions** may then be **transcoded** into Java beans or EJBs for optimizing web services).

**As to Claim 23**, Lai anticipates method of claim 17, further comprising processing said at least one accelerator output file by executing said code (as stated in par. 0892, lines 1-15 and par. 0729, lines 4-9, J2EE-compliant applications web server provides Web and EJB containers to develop and **execute** Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML-related Java APIs to develop XML-based **transforming** and **Web Services** through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be **executed** on the **mobile platform** devices. **Web Services** requester may start with one single



coarse-grained Web Service, where different services are **invoked** by passing **parameters** in the Web Service call either an XML-RPC or an XML message).

**As to Claim 37**, Lai anticipates web services program of claim 36, wherein said method further comprises transmitting at least a subset of said output data to said mobile device (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager **reformats** the **data** and **transcode** into **HTML**, or **WML** using XSL if applicable. This enables the Service Requester (**user**) to view in a **format** that is displayable on any **mobile device or browser**).

**As to Claim 38**, Lai anticipates web services program of claim 27, wherein said method further comprises receiving input data from said mobile device and invoking said web service by transmitting said input data to said web service (as stated in par. 0967, lines 1-19, par. 0892, lines 1-15 and par. 0729, lines 4-9, client requests for the Web Services start with **inputting** data as User ID and password, are represented in SOAP messages, which are carried over HTTPS. SOAP messages may also be carried over other data transport, such as SMTP or FTP. J2EE-compliant applications web server provides Web and EJB containers to develop and **execute** Java servlets and EJBs. It also supports session, state, and connection **pooling** for transaction **processing**. JAX, Java API for XML Pack is a bundle of XML-related Java APIs to develop XML-based **transforming** and **Web Services** through JAXP, JAXB, JAXM, JAXR, and JAX-

RPC modules to be **executed** on the **mobile platform** devices. **Web Services** requester may start with one single coarse-grained Web Service, where different services are **invoked** by passing **parameters** in the Web Service call either an XML-RPC or an XML message).

**As to Claim 39**, Lai anticipates web services program of claim 38, wherein said method further comprises receiving output data from said web service in response to said invoking step and transmitting said output data to said mobile device (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager **reformats** the **data received** from the **web services** which is **invoked** and **transcoded** into **HTML**, or **WML** using XSL if applicable. This enables the Service Requester (**user**) to view in a **format** that is displayable on any **mobile device or browser**).

**As to Claim 40**, Lai anticipates web services program of claim 39, wherein said method further comprises the step of detecting changes to said output data from said web service in response to said invoking step and transmitting said changes to said mobile device (as stated in par. 0240, lines 1-7, web service providers use different **messaging protocols** and data **formats**, they use SOAP or ebXML messaging to exchange **documents or instructions**. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML

structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems. Thus, the integration effort may be lower and reusable for other web services, and may not require all web service providers to use the same vendor solution or to adopt a proprietary data format. This enables any changes from back-end systems to be transmitted back to Service Requester (**user**) to view in a **format** that is displayable on any **mobile device or browser**).

**As to Claim 43**, Lai anticipates web services program of claim 42, wherein said third instructions comprise instructions for receiving said input data from said mobile device and transmitting said input data to said web service. (as stated in par. 0967, lines 1-19, par. 0892, lines 1-15 and par. 0729, lines 4-9, client requests for the Web Services with **entering** data, User ID and password, are represented in SOAP messages, which are carried over HTTPS. SOAP messages may also be carried over other data transport, such as SMTP or FTP. J2EE-compliant applications web server provides Web and EJB containers to develop and **execute** Java servlets and EJBs. It also supports session, state, and connection **pooling** for transaction **processing**. JAX, Java API for XML Pack is a bundle of XML-related Java APIs to develop XML-based **transforming** and **Web Services** through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be **executed** on the **mobile platform** devices. **Web Services** requester may start with one single coarse-grained Web Service, where different services are

*invoked* by passing *parameters* in the Web Service call either an XML-RPC or an XML message).

**As to Claim 46**, Lai anticipates web services program of claim 45, wherein said fifth instructions comprise instructions for receiving said output data from said web services and for transmitting said output data to said mobile device (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager *reformats* the *data received* from the *web services* which is *invoked* and *transcoded* into *HTML*, or *WML* using XSL if applicable. This enables the Service Requester (*user*) to view in a *format* that is displayable on any *mobile device or browser*).

**As to Claim 47**, Lai anticipates web services program of claim 46, wherein said method further comprises the step of detecting changes to said output data from said web service in response to said invoking step and transmitting said changes to said mobile device (as stated in par. 0240, lines 1-7, web service providers use different *messaging protocols* and data *formats*, they use SOAP or ebXML messaging to exchange *documents or instructions*. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems. Thus, the integration effort

may be lower and reusable for other web services, and may not require all web service providers to use the same vendor solution or to adopt a proprietary data format. This enables any changes from back-end systems to be transmitted back to Service Requester (*user*) to view in a *format* that is displayable on any *mobile device or browser*).

### **Conclusion**

3. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US Patent No. 6732167 to Swartz, Stephen J. et al., US Pat No. 6934756 to Maes, Stephane H., US. Publication No. 20020174178 to Stawikowski, Jean-Marie, US Publication No. 20050038867 to Henderson, Roderick C. et al., and US Publication No. 20060036682 to Fletcher, James C. et al., are cited for reference but not taken into consideration.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MUKTESH G. GUPTA whose telephone number is (571)270-5011. The examiner can normally be reached on Monday-Friday, 8:00 a.m. -5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Taghi T. Arani can be reached on 571-272-3787. The fax

Art Unit: 4121

phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MG

/Taghi T. Arani/  
Supervisory Patent Examiner, Art Unit 4121  
12/5/2007